

### *Amendments to the Claims*

The listing of claims will replace all prior versions, and listings of claims in the application.

1. (Canceled)
2. (Canceled)
3. (Currently Amended) A method for predicting the physical characteristics of an electronic design before gate-level implementation, the method comprising:
  - optimizing a network of logic building blocks logically and physically using placement based information to create an ~~an-accurate~~ model of the electronic design; and
  - passing optimized design information associated with the ~~accurate~~ model to gate-level implementation tools to achieve predictable results at gate-level implementation of the electronic design.
4. (Previously Presented) The method of claim 3, wherein each logic building block represents a logic structure of a fabricated electronic design and has performance data derived from placement based information for a plurality of different physical implementations of the logic structure.
5. (Previously Presented) The method of claim 3, wherein performance data of at least one logic building block is stored in a library.
6. (Previously Presented) The method of claim 3, wherein performance data of at least one logic building block is generated on-the-fly at run time.
7. (Previously Presented) The method of claim 3, wherein optimizing the network of logic building blocks includes concurrent synthesis and placement of the logic building blocks.

8. (Previously Presented) The method of claim 3, wherein optimizing the network of logic building blocks includes floorplanning.
9. (Previously Presented) The method of claim 3, wherein optimizing the network of logic building blocks includes routing estimation.
10. (Previously Presented) The method of claim 3, wherein optimizing the network of logic building blocks includes partitioning.
11. (Previously Presented) The method of claim 3, wherein optimizing the network of logic building blocks includes pin assignment.
12. (Previously Presented) The method of claim 3, wherein a design hand-off using the optimized design information enables the electronic design to be implemented to meet design requirements in a single pass through a gate-level physical implementation process.
13. (Previously Presented) The method of claim 3, further comprising:
  - receiving a description of the electronic design; and
  - synthesizing a description of the electronic design into the network of logic building blocks that are optimized, wherein the synthesizing preserves bus structures of the electronic design.
14. (Currently Amended) A method for designing an electronic design, the method comprising:
  - creating a virtual prototype using placement based information to model the electronic design thereby enabling design optimization before detail physical implementation; and
  - deriving a solution for design convergence based on data resulting from the design optimization.

15. (Previously Presented) The method of claim 14, wherein creating a virtual prototype further comprises:
  - concurrently optimizing logical and physical implementations of the electronic design using placement-based information.
16. (Previously Presented) A method for abstracting a model of an electronic design into an efficient network for optimization, the method comprising:
  - mapping the model into logic building blocks thereby creating a network of logic building blocks, a number of the logic building blocks having a logic structure including a plurality of gates thereby providing a higher level of abstraction than gates; and
  - optimizing the network of logic building blocks, each logic building block having performance data based on placed and routed implementations of that logic building block.
17. (Previously Presented) The method of claim 16, wherein mapping the model into logic building blocks preserves bus structures of the electronic design.
18. (Previously Presented) The method of claim 16, wherein each logic building block represents a logic structure of a fabricated electronic design, and has performance data derived from placement based information for a plurality of different physical implementations of the logic structure.
19. (Previously Presented) The method of claim 16, wherein performance data of at least one logic building block is stored in a library.
20. (Previously Presented) The method of claim 16, wherein performance data of at least one logic building block is generated on-the-fly at run time.
21. (Previously Presented) The method of claim 16, wherein optimizing the network of logic building blocks includes concurrent synthesis and placement of the logic building blocks.

22. (Previously Presented) The method of claim 16, wherein optimizing the network of logic building blocks includes floorplanning.
23. (Previously Presented) The method of claim 16, wherein optimizing the network of logic building blocks includes routing estimation.
24. (Previously Presented) The method of claim 16, wherein optimizing the network of logic building blocks includes partitioning.
25. (Previously Presented) The method of claim 16, wherein optimizing the network of logic building blocks includes pin assignment.
26. (Previously Presented) The method of claim 16, wherein optimizing the network of logic building blocks includes separation of data path and control logic.
27. (Previously Presented) The method of claim 16, wherein the optimized design data includes floorplan information.
28. (Previously Presented) The method of claim 16, wherein optimizing the network of logic building blocks produces optimized design data that includes one or more of timing information, routing information, placement information, netlist information, placement based wire load information, and pin assignment information.
29. (Previously Presented) The method of claim 16, wherein optimizing the network of logic building blocks produces optimized design data that enables the electronic design to be implemented to meet the design requirements in a single pass through a gate-level implementation process.
30. (Previously Presented) The method of claim 16, wherein the model is a Register-Transfer-Level description.

31. (Previously Presented) The method of claim 16, further comprising:

- clustering multiple logical building blocks into partitions thereby yielding a partition level abstraction of the electronic design;
- creating a model for each partition; and
- optimizing additional levels of the electronic design using the partition models thereby enabling hierarchical optimization without reanalyzing partition level details.

32. (Currently Amended) A method for characterizing logic building blocks, the method comprising:

- creating physical implementations of a logic building block, the logic building block having a logic structure including a plurality of gates thereby providing a higher level of abstraction than gates; and
- monitoring area and placement based performance data of each physical implementation based on a number of selected input parameter sets.

33. (Previously Presented) The method of claim 32, further comprising:

- for each selected input parameter set, storing the resulting area and performance data thereby defining a performance envelope of the logic building block.

34. (Previously Presented) The method of claim 32, further comprising:

- storing the logic building block in a library, the library capable of supplying a network of logic building blocks for representing the electronic design.

35. (Currently Amended) An electronic design computer program product encoded on one or more computer readable mediums, the product comprising:

- a library of logic structures, each logic structure having performance data based on placed and routed implementations of that logic building structure, the performance data being organized according to characteristics of the implementations; and

- a plurality of modules for optimizing a network of a number of the logic structures using placement based information to create an accurate model of the

electronic design thereby enabling optimized design information associated with the accurate model to be passed to gate-level implementation tools to achieve predictable results at gate-level implementation of the electronic design.

36. (Previously Presented) The computer program product of claim 35, wherein the performance data for each implementation quantifies a relationship between a timing delay and an output load of the implementation.

37. (Previously Presented) The computer program product of claim 35, wherein the performance data includes performance data variations for combinations of topology, implementation type, bit width, output driver size, and output load.